REMARKS

Claims 17-21, 23-26 and 28-29 are pending in the above-referenced patent application. Claims 17 and 26 are independent.

The examiner maintained the rejections of claims 17-29 under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 5,663,012 to Shimizu et al., in view of the reference "Programming, Compiling and Executing Partially-Ordered Instruction Streams on Scalable Shared-Memory Multiprocessors" by D. K. Probst (hereinafter Probst).

The examiner also maintained the rejections of claims 17-21 and 25-26 under 35 U.S.C. §103(a) over the reference Computer Systems Design and Architecture by V. P. Heuring and H. F. Jordan (hereinafter Heuring), in view of Probst. In addition, the examiner maintained the rejections of claims 22-24 and 27-29 under 35 U.S.C. §103(a) as being unpatentable over Heuring in view of Probst and in view of Shimizu.

Applicant's independent claim 17 recites "a local register instruction that loads one or more bytes, specified by a field of the instruction representing a mask, within a local destination register with a shifted value of another operand." Thus, a mask specified by a field of applicant's instruction causes bytes of a destination register to be selectively loaded with source data. For example, as described in the originally filed application, "[the mask] 0101 loads the 1st and 3rd bytes [of the destination register] while the other bytes remain unchanged" (page 11, line 27).

As explained in applicant's Amendment in Reply to Action of November 28, 2005, Shimizu describes a data processor and instructions that are capable of inserting and extracting data to and from optional bit area of a register, and to a control circuit therefor (col. 1, lines 11-15). Specifically, Shimizu describes two types of instructions; the GETxx instructions, and the PUTxx instructions. Every GETxx instruction, when executed on Shimizu's processor, causes a specific predetermined byte of the source operand to be placed into the lowest portion of a destination operand (FIG. 9, col. 4, line 49 to col. 5, line 14). For example, the instruction GETB0 extracts the first byte from the head of the source operand and places it in the lowest byte of the destination operand (FIG. 9A, FIG. 14, and col. 4, lines 53-57). In a similar vein,

execution of one of Shimizu's various PUTxx instructions causes a portion of a source operand to be placed into a specific predetermined byte position of a destination operand (FIG. 10, col. 5, lines 15-40). For example, execution of the instruction PUTB0 causes the lowest byte of the source operand to be placed into the first byte (most significant byte) of the destination operand (FIG. 10A, FIG. 17, col. 5, lines 21-24, and col. 10, line 50 to col. 11, line 5).

However, Shimizu's instructions do not describe masks to specify the particular locations of the source or destination operands. Indeed, none of the fields of any of the GETxx and/or PUTxx instructions, as shown in FIG. 7 and discussed in col. 4, lines 11-48 of Shimizu, includes a field representing a mask that specifies one or more bytes into which data from a source operand is to be loaded. If it is desired to load data from different locations of the source, or into different bytes of the destination register, it is necessary to execute different GETxx or PUTxx instructions to accomplish that.

In response to applicant's arguments in the Amendment in Reply, the examiner stated in the Final Action:

> ***Response to Arguments***
>
> 33. Applicant's arguments filed 28 February 2006 have been fully considered but they are not persuasive. Applicant argues in essence on pages 7-10 ...nowhere does Shimizu describe any instruction that uses an instruction filed, representing a mask, to specify which bytes of the destination operand to load data into. Rather, as explained above, to place a source data into a specific desired byte of the destination register, it is necessary to execute that GETxx or PUTxx instruction that would cause source data to be placed into that desired byte of the destination register...
>
> 34. This has not been found persuasive. A mask is a "pattern of characters, bits, or bytes used to control the elimination or retention of another pattern of characters, bits, or bytes (www.dictionary.com "mask" ©2000)." As admitted in Applicant's arguments, the GETxx instruction "...causes a specific predetermined byte of the source operand to be placed into the lowest portion of a destination operand..." and the PUTxx instruction "...causes a portion of a source operand to be placed into a specific predetermined byte position of a destination operand...". In essence, both instructions control which portion of bits in a portion of the destination operand is eliminated and replaced with a specific byte of a source operand. The instructions function exactly as the definition of a mask. With regards to the field of the instruction, the op-code field representing which instruction it is meets this limitation. Also, with the PUTxx instruction, the source operand field meets the limitation. (Office Action, page 13, paragraphs 33-34)

Applicant respectfully disagrees with the examiner interpretation of applicant's independent claim 17 and with the characterization of Shimizu.

As explained in MPEP 2111.01:

> **II. "PLAIN MEANING" REFERS TO THE ORDINARY AND CUSTOMARY MEANING GIVEN TO THE TERM BY THOSE OF ORDINARY SKILL IN THE ART**
>
> **>"[T]he ordinary and customary meaning of a claim term is the meaning that the term would have to a person of ordinary skill in the art in question at the time of the invention, *i.e.*, as of the effective filing date of the patent application." ... It is the use of the words in the context of the written description and customarily by those skilled in the relevant art that accurately reflects both the "ordinary" and the "customary" meaning of the terms in the claims. *Ferguson Beauregard/Logic Controls v. Mega Systems*, 350 F.3d 1327, 1338, 69 USPQ2d 1001, 1009 ...

The ordinary and customary meaning of the term "mask" recited in applicant's independent claim 1 is of a pattern of bits or bytes used to control the elimination or retention of another patterns of characters and/or bits (this is consistent with the definition identified by the examiner from *dictionary.com*). As further described in applicant's originally filed application:

> **The "Byte_ld_enables" field represents a 4-bit mask that specifies which byte(s) are affected by the instruction. Each set bit enables the corresponding byte of the destination operand longword to be loaded or cleared. There must be at least 1 set bit in the mask. For example, 0101 loads the 1st and 3rd bytes while the other bytes remain unchanged. (Page 11, lines 23-27).**

Thus, the ordinary and customary meaning of "mask," which is consistent with the way applicant has used the term, does not include instructions that are capable, without using a specified pattern of bits or bytes, to control the elimination or retention of characters/bits. Indeed, applicant's originally filed disclosure makes it clear that applicant intended the term "mask" to have its ordinary meaning, namely, of a pattern that is used, typically in conjunction with the execution of an instruction, to select bytes of an operand.

As explained above with respect to Shimizu, while it is true that execution of a particular GETxx or PUTxx instruction does cause specific byte locations of a source to be placed into specific byte locations of a destination, none of the GETxx and/or PUTxx instruction described in Shimizu uses a mask to select the specific byte locations of the operands. To construe any of the GETxx and PUTxx instructions as masks is to give the term "mask" a meaning that goes beyond the ordinary and customary meaning of the term, and is contrary to what was intended by applicant. Thus, examiner's interpretation of the word "mask" is unreasonably and impermissibly overly broad.

Likewise, to construe instructions' op-code fields as mask fields (as suggested by the examiner in the Office Action) is to give "specified by a field of the instruction representing a mask," as recited in applicant's independent claim 17, an unreasonably broad construction, which examiner is not allowed to do.

Accordingly, Shimizu neither discloses nor suggest at least the feature of "a local register instruction that loads one or more bytes, specified by a field of the instruction representing a mask, within a local destination register with a shifted value of another operand," as required by applicant's independent claim 17.

With respect to the other references cited by the examiner, the examiner admitted that:

> 31. Regarding to claims 22-24 and 27-29, Heuring in view of Probst have not taught
>
> a. Wherein the local register instruction comprises a field representing a mask that specifies which byte or bytes of the destination register are affected (Applicant's claim 22);
>
> ... (Office Action, Pages 10-11, Paragraph 30)

(Applicant notes that claim 22 was cancelled in the previous Amendment in Reply, and it is independent claim 17 that now recites the above-noted feature.)

Thus, because none of Shimizu, Heuring, and Probst discloses or suggests, alone or in combination, at least the feature of "a local register instruction that loads one or more bytes, specified by a field of the instruction representing a mask, within a local destination register with

a shifted value of another operand," applicant's independent claim 17 is therefore patentable over the cited art.

Claims 18-21, 23 and 24 depend from independent claim 17 and are therefore patentable for at least the same reasons as claim 17.

Independent claim 26 describes an apparatus featuring "a command that causes the ALU to load one or more bytes, specified by a field of the command representing a mask, within a destination register of a selected microengine with a shifted value of another one or more bytes of a source register." For similar reasons as those provided with respect to independent claim 17, at least this feature is not disclosed by the art. Claim 28-29, which depends from claim 26, are patentable for at least the same reasons as claim 26.

It is believed that all the rejections and/or objections raised by the examiner have been addressed.

In view of the foregoing, applicant respectfully submits that the application is in condition for allowance and such action is respectfully requested at the examiner's earliest convenience.

All of the dependent claims are patentable for at least the reasons for which the claims on which they depend are patentable.

Canceled claims, if any, have been canceled without prejudice or disclaimer.

Any circumstance in which the applicant has (a) addressed certain comments of the examiner does not mean that the applicant concedes other comments of the examiner, (b) made arguments for the patentability of some claims does not mean that there are not other good reasons for patentability of those claims and other claims, or (c) amended or canceled a claim does not mean that the applicant concedes any of the examiner's positions with respect to that claim or other claims.

No fee is believed due. Please apply any other charges or credits to deposit account 06-1050, referencing attorney docket 10559-320001.

Respectfully submitted,

Date: July 14, 2006

Ido Rabinovitch
Attorney for Intel Corporation
Reg. No. L0080

PTO Customer No. 20985
Fish & Richardson P.C.
Telephone: (617) 542-5070
Facsimile: (617) 542-8906

21378487.doc